Assignment 2. Linear Regression, Basis Expansion and Feature Selection

Due Date: October 30, 11:30 pm

Late submission

If you submit the assignment after the deadline, the following penalty is applied:

- ◇ 10% penalty if the submission is before November 6, 11:30 pm (if the mark before applying the penalty is 78 out of 100, after applying the penalty it is 78 - 7.8 = 70.2 out of 100);
- ◇ 50% penalty if the submission is after November 6, 11:30 pm and before December 8, 11:30 pm.

Description:

In this assignment you are required to experiment with linear regression, basis expansion and feature selection. You will use the Boston housing data set (available in scikit-learn; use "load_boston"). Feature selection refers to selecting a subset of the features to be used in the prediction.

In some cases, selecting just a subset of features might improve the prediction. Another situation when we seek to use only a subset of features is when we want to determine a smaller subset of features that affect the prediction the most.

One approach for feature selection is forward-stepwise selection. This is a greedy algorithm that starts with an empty set and grows it gradually by selecting at each step the feature that increases the performance of the predictor the most. In other words, if the selected subset at some stage is S, then a new feature is added to S as follows. Each of the unselected features is assigned a score and then the feature with higher score is chosen to be added to S. This procedure is repeated until S reaches the desired number of features.

One method to assign a score to some feature F is by measuring the performance of the predictor obtained if the feature is added to S. You will use K-fold cross-validation error as a measure of the performance. In other words, for each feature F, you will temporarily include it in S, and compute the K-fold cross-validation error. **You have to include these cross-validation errors in your report.** After doing this for all unselected features, select the feature that generated the lowest K-fold cross-validation error and add it to S[1].

In this assignment you will implement this method and experiment with it for least squares regression on the Boston housing data set. In order to see the result for each

[1] For more details and refinements of this technique and for other subset selection methods, refer to [1, Section 3.3].

possible size k of the selected subset of features, it is sufficient to run the greedy algorithm until S reaches the maximum size, i.e. 13. Additionally, for each selected subset S, you have to train the model and record the test error.

Next, for each selected subset S, you have to use basis expansion (augment or replace the set of features with some functions of them). Consider at least two models with basis expansion and select the best one using K-fold cross-validation. **You have to include in your report, the cross-validation errors for all models you consider.** The choice of the basis functions to use is yours. The goal is to obtain an improvement in performance. You will have to try more basis functions, if necessary, until at least one of your models with basis expansion achieves a cross-validation error smaller than the model without basis expansion.

At the beginning you have to split the data into the training and test sets. **Whenever you use randomization in your code, use the number formed of the last 4 digits of your student ID, as seed for the pseudo number generator.**

You have to write a report to present your results and their discussion. You have to specify the value of K that you choose. For each $k, 1 \leq k \leq 13$, you have to specify the subset of features that have been selected and the parameter vector for the k-feature model chosen (without basis expansion), and for the model with basis expansion. For the latter model, specify the basis expansion as well. Organize well these results.

The report should contain a plot of the cross-validation errors and of test errors for all 26 models that you have built, versus k (the size of the subset of features). Discuss the relation between the cross-validation error and the test error. (For each model, which one is larger? Is this relation consistent for all models? etc.). Among all models without basis expansion, which one has the smallest cross-validation error? Does the same model have the smallest test error? Answer the same questions for the models with basis expansion.

Also specify what guided you in choosing the basis functions in your trials. Additionally, specify all the sources that you have used for inspiration.

Besides the report, you have to submit your numpy code. The code has to be modular. Write a function for each of the main tasks. Also, write a function for each task that is executed multiple times (e.g, to compute the average error, to compute the K-fold cross-validation error for a model, etc). The code should include instructive comments.

Submission Instructions:

- Please submit the report in pdf format, the python file (with extension ".py") containing your code, and a short demo video. The video should be 1 min or less. In the video, you should scroll down your code, show that it runs and that it outputs the results for each part of the assignment. Submit the files in the Assignments Box on Avenue.
- More details about the steps to take for doing the assignment are given in the appendix (on the following two pages).

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd Ed., Springer, 2009 (ISBN 9780387848570), available for free download at https://web.stanford.edu/ hastie/ElemStatLearn/

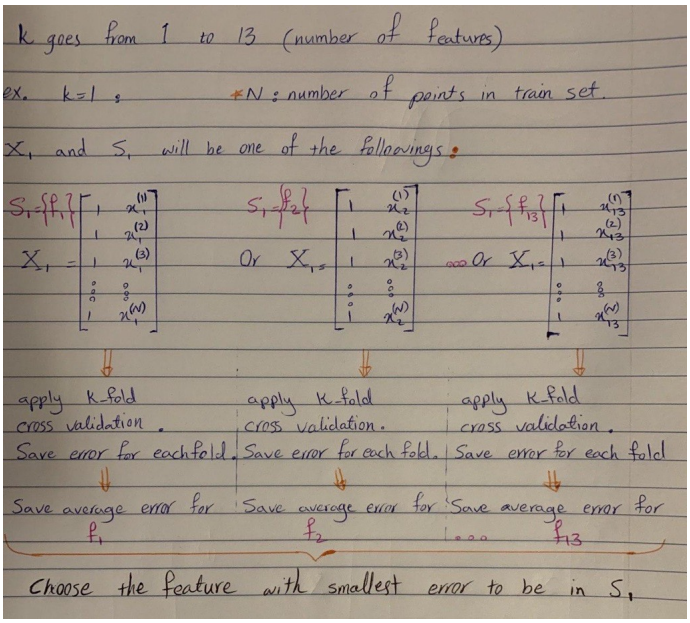
1 Appendix. Detailed Steps for Assignment 2

Prepared by Sara Zendehboodi

1.1 What to Do in the First Part of Assignment 2

In this part, you should select the most effective features among the 13 features. Consider k to be the number of steps (or the number of features is set S) in each iteration. Then k goes from 1 to 13. At the begining, split the whole data to test and train (use the instructions in the assignment). Then set the test part aside for future use. Then use the train part to train the models and find the best features.

For example, in $k = 1$, you need to find the best feature that gives the smallest error. Therefore, $|S_{(1)}| = 1$ and $|X_{train}| = N_{train} \times 1$. Note that you can add the dummy 1 to make the computations easier.



Then, in the next iteration, $k = 2$, you should find the next effective feature among the remaining 12. As an example, lets say $f_{(3)}$ was chosen in the previous iteration. Therefore, $S_{(1)} = \{f_{(3)}\}$. Now $S_{(2)}$ will have $f_{(3)}$ in it, plus a new feature that you are going to find in this iteration. So you should try $S_{(2)} = \{f_{(3)}, f_{(1)}\}$, $S_{(2)} = \{f_{(3)}, f_{(2)}\}$, $S_{(2)} = \{f_{(3)}, f_{(4)}\}$, $S_{(2)} = \{f_{(3)}, f_{(5)}\}$, up to $S_{(2)} = \{f_{(3)}, f_{(13)}\}$, to find the one with smallest error and save it in $S_{(2)}$.

You do the same process until you get to $S_{(13)}$ which is the complete set of features.

Whenever you want to measure the performance of a set S, you should apply cross validation to that set. To find the smallest error, you should use K-fold cross validation, which I will discuss in the next section.

Note that after finding the best features in each set (for each k), you have to train the model using train data only using features in that set. Then you can compute the predictions and the test error (in each k) for the test data you put aside at the begining.

2 Using K-fold Cross-validation in the Assignment

To use K-fold cross validation, follow the instruction given in Topic 3.5 in course notes. Note that α here is the feature to be found for each k.

You must give the train data to the K-fold function. For value of K, number of splits, the default is usually 5. Using "sklearn.model_selection.KFold()", you can split the train data into K folds. The way this function gives you the splits, is as follows:

1) Set $n.splits$ as the number of folds you want. Lets say 5.

2) Then give the whole train data (with only the features that you should consider in each S, each iteration inside k) as X, and the targets as Y. Then function KFold gives you 5 models for the split of the data into train/test. Note that you will get the indexes corresponding to train and test in 5 lines. The test set here, works as the validation set. This is equivalent to spliting the data into 5 folds, and each time puting 1 fold aside as the test and using the rest as the train.

3) Then for each line of index, use the train part to train the model and use the test part to test your model and find the test error. This is the error you should save and after K-fold process was finished for the 5 lines of index, you should take the average of 5 errors you found, and set it as the error of using the specific features.

If you have questions about the assignments, please attend the TA office hours using teams. If you could not attend, please leave your questions in the channels for TAs and we will respond as soon as we can.